



TITLE:

近似Groebner基底の逐次算法に向けて (Computer Algebra : Design of Algorithms, Implementations and Applications)

AUTHOR(S):

長坂, 耕作

CITATION:

長坂, 耕作. 近似Groebner基底の逐次算法に向けて (Computer Algebra : Design of Algorithms, Implementations and Applications). 数理解析研究所講究録 2012, 1815: 70-78

ISSUE DATE:

2012-10

URL:

<http://hdl.handle.net/2433/194570>

RIGHT:

近似 Groebner 基底の逐次算法に向けて

長坂耕作

KOSAKU NAGASAKA

神戸大学人間発達環境学研究科

GRADUATE SCHOOL OF HUMAN DEVELOPMENT AND ENVIRONMENT, KOBE UNIVERSITY*

1 はじめに

本講演では、近似 Gröbner 基底の計算を RREF と STLS を活用して安定化させる取り組み [7] に基づいた、計算結果である構造化 Gröbner 基底 (本稿で後述) に理論的な裏付けを保証することへの最新の成果を報告します。まず、近似 Gröbner 基底については、これまでに多くの研究論文等 [13, 14, 16, 17, 5, 15, 11, 12, 10, 8, 3] が発表されています。ただし、これらはその問題設定 (ないしはアプローチ) に大きな違いがあり、それを改めて確認しておきます。以下は、佐々木と加古 [11] による問題のクラス分けですが、本講演で扱うのは第二種になります。

第一種の近似 Gröbner 基底

厳密な係数を持つ多項式が与えられ、それを生成系とするイデアルの Gröbner 基底を数値的な演算 (例えば浮動小数点数) で求めるものを指します。例えば, Floating point gröbner bases. Shirayanagi, K. (1996) などが、これに分類されます。

第二種の近似 Gröbner 基底

先天的な誤差を持つ不正確な係数を持つ多項式が与えられ、それを生成系とするイデアルの Gröbner 基底を求めるものを指します。利用する演算の種類は問いませんが、先天的な誤差への対応が不可欠となります。例えば, Computing floating-point gröbner bases stably. Sasaki & Kako (2007) などが、これに分類されます。

第二種の問題に対する本講演の基本的な考え方は、様々な影響によって発生した先天的な誤差をなんとか排除したい、というものです。例えば、以下の多項式で生成されるイデアルの Gröbner 基底を既知の方法で計算した場合、係数が若干変化するなどの違いはあっても、以下の G を基底としえ得ることが可能です。これは、生成系を構成する多項式の性質により、多くのアルゴリズムに対して数値的に安定しており先天的な誤差の影響が少ないためです。

$$\tilde{F} = \{2.000005x + 3.000001y, 0.999999xy - 2.000003\} \Rightarrow G = \{1.0x + 1.5y, 1.0y^2 + 1.33334\}.$$

ところが、次のような多項式で生成されるイデアルの場合を考えます。これは、上の生成系に最初の多項式の 1.2 倍と最後の多項式の 0.5 倍の差において、係数部を適当な桁数で丸めた要素を追加したものです。

$$\tilde{F} = \{2.000005x + 3.000001y, 0.999999xy - 2.000003, -0.49995xy + 2.4001x + 3.59999y + 1.00001\}.$$

*nagasaka@main.h.kobe-u.ac.jp

この入力に対して Gröbner 基底を既知の方法で計算した場合、アルゴリズムにも依存しますが、生成系を構成する多項式の性質により、数値的に安定しないため結果が大きく変わってしまい、 $G = \{1\}$ のような計算結果になってしまうことがあります。注意しなければならないのは、先天的な誤差が含まれ得ると仮定しなければ、イデアルが環全体と等しくなることは、数学的には正しい結果です。ただ、多くの場合、基底を計算する状況下でイデアルが環全体と等しくなることは、計算を行うものにとっては情報量が少なく望ましい結果ではないと考えられます。そこで、入力多項式の係数を摂動させることによって、願わくば、先天的な誤差をなるべく排除することにより、数値的に安定した系の計算に置き換えたい、というのが本講演の立場です。例えば、以下のように摂動を行うことで、状況によってはより望ましい結果である $G = \{1.0x + 1.49996y, 1.0y^2 + 1.3334\}$ を得ることができます。

$$\tilde{F} = \{2.00005x + 2.99997y, 0.999981xy - 2.00000, -0.499986xy + 2.40006x + 3.60001y + 1.00001\}.$$

この摂動が真に先天的な誤差を排除した結果であるかの保証を行うことは困難ですが、計算の手続きを数学的に規定することで、計算を行うものにより多くの情報を与え、対象となっている生成系のより深い洞察を可能にすることができると考えられます。

2 構造化 Gröbner 基底

本講演の目的である構造化 Gröbner 基底の定義に先だって、Gröbner 基底を行列計算で求める方法について説明しておきます。一般に、次数に上限を設けた場合や許容する単項式に (有限の) 集合制約を設けた場合、多項式環を有限次元の加群と見做すことができ、その計算を行列演算に置き換えることが可能となります。例えば、次の生成系 F から全次数辞書式順序の Gröbner 基底 G を求めることを考えます。

$$F = \{2x + 3y, xy - 2\} \Rightarrow G = \left\{x + \frac{3y}{2}, y^2 + \frac{4}{3}\right\}.$$

生成元である $2x + 3y$ と $xy - 2$ の単項式倍の係数ベクトル (左側が項順序の全次数辞書式順序) を並べた次のような行列 $\mathcal{M}_T(F)$ を構成します。この行列の行空間は、単項式を $\{x^3, x^2y, xy^2, y^3, x^2, xy, y^2, x, y, 1\}$ に制約した多項式環に対応する加群と、 F によって生成されるイデアルとの共通部分に同型となります。

$$\mathcal{M}_T(F) = \begin{pmatrix} 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 \end{pmatrix}.$$

単項式の制約集合 $\{x^3, x^2y, xy^2, y^3, x^2, xy, y^2, x, y, 1\}$ が、ある一定の条件を満たせば、行列 $\mathcal{M}_T(F)$ の行空間に F によって生成されるイデアルの全次数辞書式順序の Gröbner 基底の基底多項式が含まれることが示されます (詳細は、[8, 6] を参考)。そこで、ガウスの消去法などでこの行列の標準形 (Reduced Row Echelon

Form, 以下 RREF) を求めると次のような結果を得られます。

$$\overline{\mathcal{M}_T(F)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{9}{2} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \frac{4}{3} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{4}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{3}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

これにより行空間の基底ベクトルが求まったわけですが, 単項式の制約集合が一定の条件を満たしていれば (この例では満たしています), 基底ベクトルに対応する多項式が, F の生成するイデアルの Gröbner 基底の基底多項式となっており, 結果として以下の Gröbner 基底 G_T が求まったことになります。なお, 冗長な基底多項式を排除すれば, 冒頭の G が得られます。

$$G_T = \left\{ x^3 - \frac{9y}{2}, yx^2 + 3y, xy^2 - 2y, y^3 + \frac{4y}{3}, x^2 + 3, xy - 2, y^2 + \frac{4}{3}, x + \frac{3y}{2} \right\}.$$

2.1 行空間の次元から潜在的な情報を捉える

前節のように, Gröbner 基底を行空間の計算に置き換えることは比較的一般的な操作になります。そこで, 先天的な誤差が混入した状況を意図的に作り, 行空間がどのように変化したかを調べてみます。ここでは, 次の多項式で生成されるイデアルの全次数辞書式順序の Gröbner 基底を計算することを例に説明していきます。単項式の制約集合は, $T = \{x^4, x^3y, x^2y^2, xy^3, y^4, x^3, x^2y, xy^2, y^3, x^2, xy, y^2\}$ とします。

$$F_{ex} = \{x^2 - 2y^2, 4x^2y + 3xy, 2x^2y + \frac{1}{2}x^2 + \frac{3}{2}xy - y^2\}.$$

この生成系に対して先ほどと同様の行列を構成すると次のようになります。この行列の階数, 即ち行空間の次元は 9 であることがわかります (誤差を考慮しない厳密な演算により求まる階数です)。

$$\mathcal{M}_T(F_{ex}) = \begin{pmatrix} 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 \\ 0 & 4 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 & 0 & \frac{1}{2} & \frac{3}{2} & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & \frac{1}{2} & \frac{3}{2} & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & \frac{1}{2} & \frac{3}{2} & -1 \end{pmatrix}, \quad \text{rank}(\mathcal{M}_T(F_{ex})) = 9.$$

一方, 生成元の一部の係数を僅かに摂動させ, それを先天的な誤差と見做した次の生成系を考えます。

$$\tilde{F}_{fp} = \{x^2 - 2y^2, 4x^2y + 3xy, 2x^2y + \frac{1}{2}x^2 + 1.500001xy - y^2\}.$$

この生成系についても、同様の単項式の制約集合を用いて行列を構成します。すると、その行空間の次元は先ほどの9ではなく、11であることがわかります。繰り返しのようになりますが、ここでいう行空間の次元、行列の階数は、厳密な意味での階数であって特異値分解などを用いた近似的な階数ではありません。なお、以下では紙面の都合から $u = 1.500001$ を用いて行列を表しています。

$$\mathcal{M}_T(\tilde{F}_{fp}) = \begin{pmatrix} 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 \\ 0 & 4 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 & 0 & \frac{1}{2} & u & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & \frac{1}{2} & u & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & \frac{1}{2} & u & -1 \end{pmatrix}, \text{rank}(\mathcal{M}_T(\tilde{F}_{fp})) = 11.$$

これらをまとめると、生成系の生成元の僅か一部の係数が $\frac{1}{2}$ から 1.500001 に摂動しただけで、行列の階数は $\text{rank}(\mathcal{M}_T(F_{ex})) = 9$ から $\text{rank}(\mathcal{M}_T(\tilde{F}_{fp})) = 11$ に変化したことになります。 $\mathcal{M}_T(\tilde{F}_{fp})$ の特異値は次のようになっており、僅かな摂動により小さい非零特異値が加わったことによる階数の変化になります。

$\{6.9034, 6.0610, 4.1684, 2.5297, 2.4445, 2.2469, 1.9732, 1.8684, 0.9618, 7.0453 \times 10^{-7}, 2.8953 \times 10^{-7}\}.$

本稿では、この性質を踏まえて、計算を行おうとした側により多くの判断材料を与えるために、先天的な誤差を含む生成系に対して、何を近似 Gröbner 基底として計算すべきか、という問題に対して、僅かな摂動で行空間の次元を減らすことができるのであれば、その摂動をした結果の生成系の Gröbner 基底を求めるべきではないか、という立場をとります。そして、これに基づいて、以下の構造化 Gröbner 基底を定義します。

定義 1 (構造化 Gröbner 基底)

G が以下の条件を満たすとき、 G を多項式集合 F に対する、許容度 $\varepsilon \in \mathbf{R}_{\geq 0}$ 、階数落ち $d \in \mathbf{Z}_{\geq 0}$ 、項集合 T 、写像族 S の構造化 Gröbner 基底 という。

1. G は以下で定義される $F_{st} = \{f_{st,1}, \dots, f_{st,k}\} \in \mathbf{C}[\vec{x}]$ の Gröbner basis 基底である。
2. F と F_{st} は S による構造化多項式集合である。即ち、写像族 S に対しパラメータ $\vec{p}_i, \vec{p}_{sti} \in \mathbf{C}^{n_i}$ が存在して、 $f_i(\vec{x}) = S_i(\vec{p}_i)$ と $f_{st,i}(\vec{x}) = S_i(\vec{p}_{sti})$ を満たす。
3. $\|\cdot\|$ をベクトルノルムとして、 $\|(\vec{p}_1 \dots \vec{p}_k) - (\vec{p}_{st1} \dots \vec{p}_{stk})\| = \varepsilon$ を満たす。
4. $\text{rank}(\mathcal{M}_T(F_{st})) = \text{rank}(\mathcal{M}_T(F)) - d.$ ◁

構造化 Gröbner 基底の例として、実際には計算が難しいものの理解し易い理想的なものとして、前述の先天的な誤差を模倣した次の生成系に対して、次の写像族 $S = \{S_1, S_2, S_3\}$ を考えます。

$$\tilde{F}_{fp} = \{x^2 - 2y^2, 4x^2y + 3xy, 2x^2y + \frac{1}{2}x^2 + 1.500001xy - y^2\}.$$

$$S_1: (p_1 \ p_2) \mapsto p_1x^2 + p_2y^2,$$

$$S_2: (p_3 \ p_4) \mapsto p_3x^2y + p_4xy,$$

$$S_3: (p_5 \ p_6 \ p_7 \ p_8) \mapsto p_5x^2y + p_6x^2 + p_7xy + p_8y^2.$$

このとき、先天的な誤差を模倣する前の生成系である次の F_{st} は、構造化 Gröbner 基底の定義における F_{st} の条件を満たしており、 $G = \{x^2 - 2y^2, y^3 + \frac{3}{8}xy\}$ は、全次数辞書式、許容度 $\varepsilon = 0.000001$ (2-norm)、階数落ち $d = 2$ 、項集合 $T = \{x^4, x^3y, x^2y^2, xy^3, y^4, x^3, x^2y, xy^2, y^3, x^2, xy, y^2\}$ の構造化 Gröbner 基底となります。

$$F_{st} = \{x^2 - 2y^2, 4x^2y + 3xy, 2x^2y + \frac{1}{2}x^2 + \frac{3}{2}xy - y^2\}.$$

2.2 構造化 Gröbner 基底の計算とその問題

実際に構造化 Gröbner 基底を計算するためには、構造を保ちながら階数を落とした行列を求める必要があります。これは、以下の SLRA という問題になります。

定義 2 (SLRA: Structured Low Rank Approximation)

Given a structure specification $S: \mathbf{R}^{n_a} \rightarrow \mathbf{R}^{m \times n}$, a parameter vector $\vec{p} \in \mathbf{R}^{n_a}$, a vector norm $\|\cdot\|$, and an integer r , $0 < r < \min\{m, n\}$, find a vector \vec{p}^* such that $\min_{\vec{p}^*} \|\vec{p} - \vec{p}^*\|$ and $\text{rank}(S(\vec{p}^*)) \leq r$.

SLRA について補足をしておくと、近似代数で良く使われる STLS (Structured Total Least Squares) は、SLRA の特別な場合 ($r = \min\{m, n\} - 1$) になります。ただし、SLRA を STLS へ帰着することも理論的には可能 (Boito 2007, p.75–77[1] や Park/Zhang/Rosen 1999[9] に記載) ですが、実際には有益なベクトルの選定を行うことが実質的に不可能なので、ほぼ困難と言えます。そして、SLRA は一般には NP 困難であって、その解法としては、局所最適解を求める「lift-and-project 法」が知られています。

以下に、これまでに研究会などで報告した、SLRA の解法を用いたアルゴリズムを再度掲載しておきます。

アルゴリズム 1 (局所最適解による構造化 Gröbner 基底の計算)

入力: 項順序 \succ , 写像族 S , 多項式集合 $F = \{f_1(\vec{x}), \dots, f_k(\vec{x})\}$.

出力: 構造化 Gröbner 基底 G , 許容度 ε , 階数落ち d , 項集合 T , 摂動後の多項式集合 F_{st} , または「失敗」

1. F の Gröbner 基底を計算 (近似か厳密) し、項集合 T を推定。
2. 行列 $\mathcal{M}_T(F) \in \mathbf{C}^{m \times n}$ の特異値 σ_i ($i = 1, \dots, r_{org}$) を計算。
3. 階数落ち d を推定 ($\sigma_{r_{org}-d+1}/\sigma_{r_{org}-d} < 10^{-2}$ なる最大の d など)。 d がなければ、「失敗」を出力。
4. SLRA の解法 (lift-and-project 法など) で、 $\text{rank}(\mathcal{M}_T(F_{st})) = \text{rank}(\mathcal{M}_T(F)) - d$ を満たす構造化多項式集合 F_{st} と許容度 ε を計算。そのような F_{st} が見つけれなければ、「失敗」を出力。
5. F_{st} の近似 Gröbner 基底 G を既知の方法で計算。 $\{G, \varepsilon, d, T, F_{st}\}$ を出力。 ◀

このアルゴリズムによる構造化 Gröbner 基底の計算には、改善すべき問題として、1) 適切な単項式の制約集合である項集合 T の決定方法が示されていない、2) この種の SLRA 問題に特化した SLRA の解法の開発、3) 許容度の点で最適な構造化 Gröbner 基底の計算方法などが挙げられます。そこで、本講演ではこのうち、適切な項集合 T の決定方法として、行列演算によって高速に Gröbner 基底を計算する F_4 アルゴリズム [2] と SLRA による手法を組み合わせたらどうかという視点から、逐次的な算法による項集合の決定について論じます。

3 逐次的な項集合の決定

構造化 Gröbner 基底の計算では、SLRA として階数落ちの行列を求める前に、Gröbner 基底の計算に必要な項集合 T を決定する必要があります。これは、既知の Gröbner 基底の計算アルゴリズムと大きく異なる

る点です。そこで、本稿では、行列計算による基底計算アルゴリズムである F_4 の手順に沿って、SLRA によって後ほど階数を落とすことを考慮に入れて、必要な項集合を決定する方法を提案します。重要なのは、我々に必要なのは基底計算に必要な項集合であって、通常のアプローチが必要とする S 多項式などを具体的に求めることが、少なくともこの段階では必要ないことです。従って、次のような方法が考えられることになります。

- QR 分解で行消去を行い、特異値でゼロ判定を行う (誤差に対応し、本来消去されるものを消去)。
- SLRA は項集合の決定後に一括して行う (各行列で SLRA を行うと、多項式の同一性が失われる)。
- 新たな項の検出に留め、毎回全ての行を再構成 (F_4 の Simplify など使わない)。

この方針で、例として全次数辞書式順序で、次の生成系の Gröbner 基底計算に必要な項集合を、sugar ストラテジ [4] に基づいて求めてみます。厳密な計算に必要な項集合でないことに留意して下さい。

$$F = \{f_1 = x^2 - 2y^2, f_2 = 4x^2y + 3xy, f_3 = 2x^2y + 0.5x^2 + 1.500001xy - y^2\}.$$

まず、上の方針に基づいて必要となる情報を、各基底候補の多項式に付与します。

$$\left(\begin{array}{ccc} \text{頭項} & \text{項集合} & \text{Sugar} \\ f_1 & x^2 & \{x^2, y^2\} & 2 \\ f_2 & x^2y & \{x^2y, xy\} & 3 \\ f_3 & x^2y & \{x^2y, x^2, xy, y^2\} & 3 \end{array} \right)$$

この状態で通常の sugar ストラテジに基づいて F_4 アルゴリズムに似た手続きを行っていきます。まず、クリティカルペア: $(f_1, f_3), (f_1, f_2)$ のうち、sugar に基づいて (f_1, f_3) の計算を行うことになります。 S 多項式は、 $y \times f_1$ と f_3 による頭項消去になるので、この計算に必要な項集合は、 $y\{x^2, y^2\} \cup \{x^2y, x^2, xy, y^2\} = \{x^2y, y^3, x^2, xy, y^2\}$ となります。これを既知の基底多項式で項簡約を行うのに必要な項集合に拡張する必要がありますが、この例では、項簡約も同じ項集合で可能なことがわかります。

このクリティカルペアによって、新たにどのような頭項を持つ基底多項式が追加されるのかを、前章と同じ性質の行列を構成して調べます。その行列は次のようになり、特異値を計算することにより、多項式を摂動させた場合に階数が 1 落ちる可能性があることが判明します。

$$M = \begin{pmatrix} 0 & 0 & 1 & 0 & -2 \\ 1 & -2 & 0 & 0 & 0 \\ 4 & 0 & 0 & 3 & 0 \\ 2 & 0 & 0.5 & 1.5 & -1 \end{pmatrix}, \text{ 特異値 : } \{1.47753, 1.03488, 0.863669, 1.978477 \times 10^{-7}\}.$$

そこで、この最小特異値を誤差と見做す閾値としてこの行列を QR 分解すると、次のように行空間の基底ベクトルが求まります。このうち、2 行目の行ベクトルは頭項として新たな項 y^3 を持っています。

$$\left(\begin{array}{ccccc} x^2y & y^3 & x^2 & xy & y^2 \\ -1.17189 & 0.341328 & -0.113776 & -0.750922 & 0.227552 \\ 0 & 0.826738 & 0 & 0.310027 & -0.0939474 \\ 0 & 0 & 0.467099 & 0 & -0.934199 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

そこで、具体的な項簡約後の多項式は不明 (多項式の摂動により変化するため) ですが、頭項として y^3 を持ち、それを再構成するのに項集合 $\{x^2y, y^3, x^2, xy, y^2\}$ が必要な基底多項式を f_4 として生成系に追加します。

$$\begin{pmatrix} f_1 & x^2 & \{x^2, y^2\} & 2 \\ f_2 & x^2y & \{x^2y, xy\} & 3 \\ f_3 & x^2y & \{x^2y, x^2, xy, y^2\} & 3 \\ f_4 & y^3 & \{x^2y, y^3, x^2, xy, y^2\} & 3 \end{pmatrix}$$

引き続き、通常のアルゴリズムに沿って手続きを引き続き行っていきます。クリティカルペアの残りは、 f_4 が追加されたことを考慮しても、 (f_1, f_2) となり、 f_1 と f_2 の S 多項式を計算することになります。S 多項式は、 $y \times f_1$ と f_2 による頭項消去になるので、この計算に必要な項集合は、 $y\{x^2, y^2\} \cup \{x^2y, xy\} = \{x^2y, y^3, xy\}$ であることがわかります。この項集合に含まれる項は f_3 による項簡約を行うことが可能なので、項簡約を行うのに必要な項集合を求めると、 $\{x^2y, y^3, xy\} \cup \{x^2y, x^2, xy, y^2\} = \{x^2y, y^3, x^2, xy, y^2\}$ となります。この項集合は先程と同じなので、同じ行列が構成され、新たな頭項は出現しないことがわかります。クリティカルペアは全て計算が終わったので、最終的に、この生成系の Gröbner 基底計算に必要な単項式の制約集合は、 $\{x^2y, y^3, x^2, xy, y^2\}$ であることが判明したことになります。

この手順をアルゴリズムとして記述すると次のようになります。

アルゴリズム 2 (単項式の制約集合)

入力: 項順序 \succ , 多項式集合 $F_{init} = \{f_1(\vec{x}), \dots, f_k(\vec{x})\}$

出力: 構造化 Gröbner 基底計算に使う項集合 \mathcal{T}

1. $F = F_{init}$, $\mathcal{T} = \{\}$, $\mathcal{T}_F = \{\text{supp}(f_i) \mid f_i \in F_{init}\}$, P : ペア集合
2. while $P \neq \phi$
 - (a) $\{i, j\} \in P$, $P = P \setminus \{i, j\}$
 - (b) $\mathcal{T}_{ij} = \text{TermUpdate}(F, \mathcal{T}_F, \{i, j\})$
 - (c) F_{init} の係数行列 $M_{\mathcal{T}_{ij}}$ を \mathcal{T}_{ij} に基づいて構成
 - (d) $M_{\mathcal{T}_{ij}}$ の QR 分解 (特異値分解に基づき結果解釈) により、新しい基底多項式を検出する。結果を $F_{new} = \{\hat{f}_1, \dots, \hat{f}_r\}$ とする。
 - (e) $F = F \cup F_{new}$, $\mathcal{T}_F = \mathcal{T}_F \cup \{\mathcal{T}_{ij}, \dots, \mathcal{T}_{ij}\}$, P の更新
3. \mathcal{T} を出力

アルゴリズム 3 (TermUpdate)

入力: $F = \{f_1(\vec{x}), \dots, f_k(\vec{x})\}$, 項集合 $\mathcal{T}_F = \{\mathcal{T}_1, \dots, \mathcal{T}_k\}$, ペア $\{i, j\}$

出力: 当該ペアの計算に必要な項集合 \mathcal{T}

1. $\mathcal{T}_{\triangleleft F} = \{\mathcal{T}_{\triangleleft i} = \{t \in \mathcal{T}_i \mid t \succeq \text{ht}(f_i)\} \mid \mathcal{T}_i \in \mathcal{T}_F\}$
 $\mathcal{T}_{\triangleright F} = \{\mathcal{T}_{\triangleright i} = \{t \in \mathcal{T}_i \mid t \prec \text{ht}(f_i)\} \mid \mathcal{T}_i \in \mathcal{T}_F\}$
2. $\mathcal{T}_{\triangleleft} = \text{lcm}(\text{ht}(f_i), \text{ht}(f_j)) / \text{ht}(f_i) \mathcal{T}_{\triangleleft i} \cup \text{lcm}(\text{ht}(f_i), \text{ht}(f_j)) / \text{ht}(f_j) \mathcal{T}_{\triangleleft j}$
 $\mathcal{T}_{\triangleright} = \text{lcm}(\text{ht}(f_i), \text{ht}(f_j)) / \text{ht}(f_i) \mathcal{T}_{\triangleright i} \cup \text{lcm}(\text{ht}(f_i), \text{ht}(f_j)) / \text{ht}(f_j) \mathcal{T}_{\triangleright j}$
3. while $\mathcal{T} \neq \phi$
 - (a) $t \in \mathcal{T}$, $\mathcal{T} = \mathcal{T} \setminus \{t\}$
 - (b) for $i \in \{1, \dots, k\}$ s.t. $\text{ht}(f_i) \mid t$

i. $t/\text{ht}(f_i)\mathcal{T}_i \setminus \mathcal{T}_a \cup \mathcal{T}_b \neq \emptyset$ ならば,

$$\mathcal{T}_a = \mathcal{T}_a \cup t/\text{ht}(f_i)\mathcal{T}_{ai}, \quad \mathcal{T}_b = \mathcal{T}_b \cup t/\text{ht}(f_i)\mathcal{T}_{bi}, \quad \mathcal{T} = \mathcal{T} \cup t/\text{ht}(f_i)\mathcal{T}_{bi}$$

4. $\mathcal{T} = \mathcal{T}_a \cup \mathcal{T}_b$ を出力

◁

このアルゴリズムの全次数辞書式順序での計算例を掲載しておきます。先天的な誤差を模倣して意図的に摂動させる前の、通常であれば決してたどり着くことのできない真の入力は次の生成系を想定します。

$$\begin{aligned} \text{生成系: } F &= \left\{ x^2 - 2y^2, 4x^2y + 3xy, 2x^2y + \frac{x^2}{2} + \frac{3xy}{2} - y^2 \right\} \\ \text{Gröbner 基底 } G &= \left\{ x^2 - 2y^2, y^3 + \frac{3xy}{8} \right\} \\ \text{必要な項集合 } \mathcal{T} &= \{x^2y, y^3, x^2, xy, y^2\} \end{aligned}$$

摂動 (誤差) を加えたアルゴリズムに入力される多項式集合として、次を考えます。

$$F = \{1.01x^2 - 2.09y^2 + 0.002, 4.03x^2y + 3.06xy, 2.04x^2y + 0.504x^2 + 1.504xy - 1.02y^2\}.$$

まず、アルゴリズムはクリティカルペア：(1, 3) に対して、その項集合 $\{x^2y, y^3, x^2, xy, y^2, y, 1\}$ を求め、構成した行列の特異値 $\{1.47461, 1.03445, 0.86911, 0.00912252\}$ とその QR 分解から、頭項 y^3 を持ち必要な項集合が $\{x^2y, y^3, y\}$ である新たな生成元を追加します。次に、クリティカルペア：(1, 2) に対して、その項集合 $\{x^2y, y^3, xy, y\}$ を求め、特異値が $\{1.1604, 0.808372\}$ となる行列を構成し QR 分解をしますが、新たな頭項を持つ基底多項式は検出されません。クリティカルペアの計算が全て終わり、結果として、項集合 $\{x^2y, y^3, x^2, xy, y^2, y, 1\} \cup \{x^2y, y^3, xy, y\} = \{x^2y, y^3, x^2, xy, y^2, y, 1\}$ が求まりました。

この項集合を用いて、アルゴリズム 1 と同じ方法で構造化 Gröbner 基底を求めると、 $\{1.0x^2 - 2.06569y^2 + 7.73225 \times 10^{-16}, 1.0y^3 + 0.362706xy\}$ が得られ、許容度は 0.0390898 となります。摂動前の厳密な結果に近づいていることがわかります。ただし、一般には、誤差の入り方は様々なため、この結果により、入力多項式の本来の姿に近づいたとは言えません。あくまでも、本講演の目的である、入力した誤差を含む生成系の解釈に付加的な情報が計算されただけに過ぎません。

4 まとめ

本講演での方法 (F_4 の手順を誤差に対応させることで項集合を計算) により、構造化 Gröbner 基底に必要な項集合の計算を行うことができました。しかしながら、この方法には QR 分解の閾値の適切性など、1) 効率化 (QR 分解と特異値計算の階数を減らす工夫)、2) QR 分解の要素のゼロ判定の理論的な議論、3) F_4 と同じく項簡約可能な基底多項式を全て使わないことが可能かの議論、4) 最後にそのまま QR 分解で構造化 Gröbner 基底を出力することの議論などの課題もあり今後の研究が必要となっています。なお、本研究の一部は科研費 (22700011) の支援で行われています。

参考文献

- [1] P. Boito. *Structured Matrix Based Methods for Approximate GCD*. Ph.D. Thesis. Department of Mathematics, University of Pisa, Italia, 2007.
- [2] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139(1-3):61–88, 1999.
- [3] J.-C. Faugère and Y. Liang. Artificial discontinuities of single-parametric gröbner bases. *J. Symb. Comput.*, (in press, 8 pages), 2010.

- [4] A. Giovini, T. Mora, G. Niesi, L. Robbiano, and C. Traverso. "one sugar cube, please" or selection strategies in the buchberger algorithm. In *Proceedings of the 1991 international symposium on Symbolic and algebraic computation*, ISSAC '91, pages 49–54, New York, NY, USA, 1991. ACM.
- [5] A. Kondratyev, H. J. Stetter, and S. Winkler. Numerical computation of gröbner bases. In *Proceedings of CASC2004 (Computer Algebra in Scientific Computing)*, pages 295–306, 2004.
- [6] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *Computer algebra (London, 1983)*, volume 162 of *Lecture Notes in Comput. Sci.*, pages 146–156. Springer, Berlin, 1983.
- [7] 長坂耕作. 近似 Groebner 基底に向けて. 研究集会 Computer Algebra - Design of Algorithms, Implementations and Applications. 京都大学数理解析研究所. 2009 年 11 月.
- [8] K. Nagasaka. A study on gröbner basis with inexact input. In *Proceedings of CASC 2009*, volume 5743 of *Lecture Notes in Comput. Sci.*, pages 247–258. Springer, Berlin, 2009.
- [9] H. Park, L. Zhang, and J. B. Rosen. Low rank approximation of a Hankel matrix by structured total least norm. *BIT*, 39(4):757–779, 1999.
- [10] T. Sasaki and F. Kako. Computing floating-point gröbner bases stably. In *Proceedings of SNC 2007*, pages 180–189. ACM, New York, 2007.
- [11] T. Sasaki and F. Kako. Floating-point gröbner basis computation with ill-conditionedness estimation. In *Proceedings of ASCM 2007*, volume 5081 of *Lecture Notes in Comput. Sci.*, pages 278–292. Springer, Berlin, 2008.
- [12] T. Sasaki and F. Kako. Term cancellations in computing floating-point gröbner bases. In *Proceedings of CASC 2010*, volume 6244 of *Lecture Notes in Comput. Sci.*, pages 220–231, Berlin, 2010. Springer.
- [13] K. Shirayanagi. An algorithm to compute floating point gröbner bases. In *Proceedings of the Maple summer workshop and symposium on Mathematical computation with Maple V : ideas and applications*, pages 95–106, Cambridge, MA, USA, 1993. Birkhauser Boston Inc.
- [14] K. Shirayanagi. Floating point gröbner bases. In *Selected papers presented at the international IMACS symposium on Symbolic computation, new trends and developments*, pages 509–528, Amsterdam, The Netherlands, The Netherlands, 1996. Elsevier Science Publishers B. V.
- [15] H. J. Stetter. Approximate gröbner bases – an impossible concept? In *Proceedings of SNC 2005 (Symbolic-Numeric Computation)*, pages 235–236, 2005.
- [16] C. Traverso and A. Zanoni. Numerical stability and stabilization of groebner basis computation. In *ISSAC 2002: Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 262–269, New York, NY, USA, 2002. ACM.
- [17] V. Weispfenning. Gröbner bases for inexact input data. In *Proceedings of CASC 2003 (Computer Algebra in Scientific Computing)*, pages 403–411, 2002.